(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－304232
（P2002－304232A）
(43)公開日　平成14年10月18日(2002.10.18)

(51)Int.Cl.⁷		識別記号	FI			テーマコード（参考）
Ｇ０６Ｆ	1/04	３０１	Ｃ０６Ｆ	1/04	３０１Ｃ	５Ｂ０１１
	1/32			9/30	３３０Ｂ	５Ｂ０３３
	9/30	３３０		9/46	３４０Ｂ	５Ｂ０７９
	9/46	３４０			３４０Ｅ	５Ｂ０９８
				1/00	３３２Ｚ	

審査請求　未請求　請求項の数15　ＯＬ　（全 16 頁）

(21)出願番号　特願2001－104560(P2001－104560)

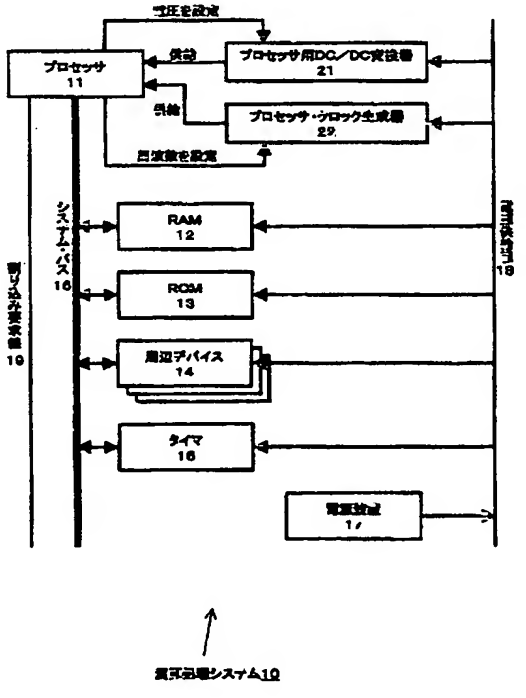
(22)出願日　平成13年４月３日(2001.4.3)

(71)出願人　000002185
ソニー株式会社
東京都品川区北品川６丁目７番35号
(72)発明者　戸川　敦之
東京都品川区北品川６丁目７番35号　ソニー株式会社内
(74)代理人　100101801
弁理士　山田　英治　（外２名）

最終頁に続く

(54)【発明の名称】　演算処理システム及び演算処理制御方法、並びに記憶媒体

(57)【要約】
【課題】　応答時間に対する要求が課せられた処理を、より低い消費電力で実行する。
【解決手段】　システムは、プロセッサに与える動作周波数と電源電圧をシステム稼動中に変化させるハードウェア機構を備える。電力マネージャは、待ち状態（キー入力やマウスのクリックなどの入出力を待機する状態）にあるときは、他の負荷を処理するのに必要充分な動作周波数と電源電圧を設定する。待ち状態から脱してから経過時間に応じて、動作周波数と電源電圧を変化させていくことで、プロセッサの消費電力を削減する。



演算処理システム10

【特許請求の範囲】
【請求項１】応答時間に対する要求が課せられた処理タスクを実行可能なプロセッサを含んだ演算処理システムであって、
前記プロセッサの稼動中の各時点において、該応答時間に対する要求を満たすために充分な前記プロセッサの動作周波数を決定する動作周波数決定手段と、
前記動作周波数決定手段による決定結果に基づく動作周波数クロックを生成して前記プロセッサに供給するプロセッサ・クロック生成手段と、を具備することを特徴とする演算処理システム。
【請求項２】前記動作周波数決定手段により決定された動作周波数で前記プロセッサを駆動させるために充分な前記プロセッサの電源電圧を決定する電源電圧決定手段と、
前記電源電圧決定手段による決定結果に基づく電源電圧を生成して前記プロセッサに供給するプロセッサ電源供給手段と、をさらに備えることを特徴とする請求項１に記載の演算処理システム。
【請求項３】前記動作周波数決定手段は、処理タスクが待ち状態を脱してからの経過時間に応じて前記プロセッサの動作周波数を決定する、ことを特徴とする請求項１に記載の演算処理システム。
【請求項４】前記動作周波数決定手段は、処理タスクが待ち状態を脱してから所定時間が経過したことに応じて前記プロセッサの動作周波数を段階的に増大させる、ことを特徴とする請求項１に記載の演算処理システム。
【請求項５】前記動作周波数決定手段は、処理タスクが待ち状態を脱してから所定時間が経過するまでの期間に前記プロセッサの動作周波数を連続的に増大させる、ことを特徴とする請求項１に記載の演算処理システム。
【請求項６】前記プロセッサは、応答時間に対する要求が課せられた処理タスクを含んだ１以上のタスクを並行して実行することが可能であり、
前記動作周波数決定手段は、応答時間に対する要求が課せられた処理タスクが待ち状態にあるときは、前記プロセッサに課された他のタスクを処理するのに充分な前記プロセッサの動作周波数を決定する、ことを特徴とする請求項１に記載の演算処理システム。
【請求項７】応答時間に対する要求が課せられた処理タスクを実行可能なプロセッサによる演算処理を制御する演算処理制御方法であって、
前記プロセッサの稼動中の各時点において、該応答時間に対する要求を満たすために充分な前記プロセッサの動作周波数を決定する動作周波数決定ステップと、
前記動作周波数決定ステップによる決定結果に基づく動作周波数クロックを生成して前記プロセッサに供給するプロセッサ・クロック生成ステップと、を具備することを特徴とする演算処理制御方法。
【請求項８】前記動作周波数決定ステップにより決定された動作周波数で前記プロセッサを駆動させるために充分な前記プロセッサの電源電圧を決定する電源電圧決定ステップと、
前記電源電圧決定ステップによる決定結果に基づく電源電圧を生成して前記プロセッサに供給するプロセッサ電源供給ステップと、をさらに備えることを特徴とする請求項７に記載の演算処理制御方法。
【請求項９】前記動作周波数決定ステップでは、処理タスクが待ち状態を脱してからの経過時間に応じて前記プロセッサの動作周波数を決定する、ことを特徴とする請求項７に記載の演算処理制御方法。
【請求項１０】前記動作周波数決定ステップでは、処理タスクが待ち状態を脱してから所定時間が経過したことに応じて前記プロセッサの動作周波数を段階的に増大させる、ことを特徴とする請求項７に記載の演算処理制御方法。
【請求項１１】前記動作周波数決定ステップでは、処理タスクが待ち状態を脱してから所定時間が経過するまでの期間に前記プロセッサの動作周波数を連続的に増大させる、ことを特徴とする請求項７に記載の演算処理制御方法。
【請求項１２】前記プロセッサは、応答時間に対する要求が課せられた処理タスクを含んだ１以上のタスクを並行して実行することが可能であり、
前記動作周波数決定ステップでは、応答時間に対する要求が課せられた処理タスクが待ち状態にあるときは、前記プロセッサに課された他のタスクを処理するのに充分な前記プロセッサの動作周波数を決定する、ことを特徴とする請求項７に記載の演算処理制御方法。
【請求項１３】応答時間に対する要求が課せられた処理タスクを実行可能なプロセッサによる演算処理の制御をコンピュータ システム上で実行するように記述されたコンピュータ ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ ソフトウェアは、
前記プロセッサの稼動中の各時点において、該応答時間に対する要求を満たすために充分な前記プロセッサの動作周波数を決定する動作周波数決定ステップと、
前記動作周波数決定ステップによる決定結果に基づく動作周波数クロックを生成して前記プロセッサに供給するプロセッサ・クロック生成ステップと、を具備することを特徴とする記憶媒体。
【請求項１４】前記動作周波数決定ステップでは、処理タスクが待ち状態を脱してから所定時間が経過するまでの期間に前記プロセッサの動作周波数を連続的に増大させる、ことを特徴とする請求項１３に記載の記憶媒体。
【請求項１５】前記プロセッサは、応答時間に対する要求が課せられた処理タスクを含んだ１以上のタスクを並行して実行することが可能であり、
前記動作周波数決定ステップでは、応答時間に対する要

求が課せられた処理タスクが待ち状態にあるときは、前記プロセッサに課された他のタスクを処理するのに充分な前記プロセッサの動作周波数を決定する、ことを特徴とする請求項１３に記載の記憶媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、１以上のタスクを同時に実行するタイプのプロセッサに関する電力削減技術に係り、特に、応答時間に対する要求が課されたタスクを含んだ１以上のタスクを実行するプロセッサに関する電力削減技術に関する。

【０００２】更に詳しくは、本発明は、動作周波数と電源電圧を変化させる機構を備えたプロセッサに関する電力削減技術に係り、ＧＵＩ（Graphical User Interface）処理のように、あらかじめ実行開始時間や処理時間が予測することができないタスクを実行時の動作周波数と電源電圧を最適に設定することにより低消費電力化を図るプロセッサに関する電力削減技術に関する。

【０００３】

【従来の技術】昨今のＬＳＩ（Large Scale Integration）技術における革新的な進歩とも相俟って、各種の情報処理機器や情報通信機器が開発され、市販されるようになってきた。この種の機器では、ＣＰＵ（Central Processing Unit）やその他のプロセッサが所定のプログラム・コードを実行することによりさまざまな処理サービスを提供するようになっている。

【０００４】他方において、情報機器に関する消費電力の削減が産業界における最重要課題の１つとされている。これは、バッテリ駆動式の情報機器においてはバッテリ持続時間の延長に関わる問題だからである。また、商用電源で無尽蔵に駆動することができる情報機器においても、資源有限という社会生態学的な観点から省電力化が推奨されている。

【０００５】情報機器内では、そのメイン・コントローラであるプロセッサの消費電力は、機器全体のそれに占める割合は高い。言い換えれば、プロセッサの省電力化は情報機器自体の省電力化にもつながる。一般には、プロセッサは、動作周波数の増大に従って演算速度が向上する一方で、その消費電力（さらには発熱量）が増大する傾向にある。

【０００６】例えば、特開平１１－１９４８４９号公報には、消費電力を無用に増加させることなく所定の処理時間に所定の処理動作を完了することができ、タスクの処理容量が変化する場合でも設定作業が簡単となるデータ処理方法及び装置について開示している。

【０００７】同公報に開示されたデータ処理装置では、マイクロコンピュータが各種の処理動作を実行する場合の処理容量と処理時間を容量記憶手段と時間記憶手段とに登録しておき、マイクロコンピュータが各種の処理動作を実行する場合に対応する処理容量及び処理時間を選出し、処理容量を処理時間で除算してマイクロコンピュータの処理速度を算出して基準クロックの周波数を可変としている。マイクロコンピュータの処理速度を処理容量と処理時間に対応して可変するので、所定の処理動作を所定の処理時間に確実に完了することができるとともに、基準クロックの周波数を最適値に設定できるので、データ処理装置における消費電力の無用な増加も防止することができる。

【０００８】しかしながら、同公報に開示されるデータ処理方法及び装置では、プロセッサの動作クロック周波数を変更するだけで消費電力の削減を図るものである。言い換えれば、動作クロック周波数の削減によって、単位時間当りの消費電力は低下するものの、各処理を完了させるための所要時間が長くなり、この結果、総電力量を削減する効果はあまり高くない。すなわち、プロセッサがアイドリング状態にあるときの消費電力量の範囲を越えず、効果として不充分である。

【０００９】また、同公報に開示されるデータ処理方法及び装置は、各処理の処理タイミングがあらかじめ定まっており、且つ、各処理を中断することなく順次処理することによってすべての処理を時間内に完了させることが可能なことを前提とするものである。このため、ある処理の実行を中断して、より緊急度が高い処理（例えばリアルタイム処理）を行わせる必要があるシステムに対しては適用することができない。

【００１０】また、特開２０００－１２２７４７号公報には、デジタル信号演算処理部にクロックを供給するクロック発生部を設けて、このクロック発生部からデジタル信号演算処理部へ供給するクロック周波数を、デジタル信号演算処理部での演算処理量に基づいて制御することによって消費電力を低減する制御装置及び方法について開示されている。

【００１１】しかしながら、同公報に開示される制御装置及び方法では、演算部の動作クロック周波数を変更するだけで消費電力の削減を図るものである。言い換えれば、動作クロック周波数の削減によって、単位時間あたりの消費電力は低下するものの、各処理を完了させるための所要時間が長くなり、この結果、電力量の削減効果は、演算部がアイドリング状態にあるときの消費電力量の範囲を越えず、効果として不充分である。

【００１２】また、同公報に開示される制御装置及び方法では、アイドリング時間が占める割合から動作周波数を算出するようになっている。ところが、アイドリング時間が占める割合は時々刻々と変化するので、その変化を予測することは極めて困難である。

【００１３】また、Takanori Okuma, Tohru Ishihara, Hiroto Yasuura共著の論文"Real-Time Task Scheduling for a Variable Voltage Processor"（IEEE 12th International Symposium on System Synthesis, November 1999）において提案されるＳＳ及びＳＤなるスケジュー

リング手法では、システムの稼動前に、タスクの実行開始時間が判っていることを前提としている。このため、ＧＵＩ（Graphical User Interface）処理のように、あらかじめ実行開始時間や処理時間が予測することができない処理に対して適用することができないという問題がある。

【００１４】また、この論文で提案されているＤＤスケジューリング手法は、ＧＵＩ処理のように明確なデッドラインを持たない処理に対しては適用することができないという問題がある。

【００１５】また、Yann-Hang Lee, C. M. Krishna共著の論文"Voltage-Clock Scaling forLow Energy Consumption in Real-time Embedded Systems"（IEEE Sixth International Conference on Real-Time Computing Systems and Applications, December 1999）において提案されている"Task based static scheduling"なる手法は、システムの稼動前にタスクの実行時間があらかじめ判っていることを前提としている。このため、ＧＵＩ処理のように、あらかじめ実行時間や処理時間が予測できない処理に対して適用することができないという問題がある。

【００１６】また、本出願人に既に譲渡されている、特願２０００－２８７８８２号明細書並びに特願２０００－２８７８８３号明細書には、動作周波数と電源電圧が可変なタイプのプロセッサを含んだ演算処理システムにおけるタスク実行の省電力スケジューリングについて提案されている。

【００１７】このうち、特願２０００－２８７８８２号明細書には、アプリケーションのリアルタイム要求に応えつつプロセッサによる電力消費を削減することができる省電力スケジューリング手法について開示されている。すなわち、プロセッサの動作周波数と電源電圧をオペレーティング・システムの制御により変化させることが可能なシステムにおいて、起動された周期リアルタイム・タスク並びに非リアルタイム・タスクを遅滞なく処理するために必要なプロセッサの動作周波数を適応的に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適なプロセッサ用電源電圧を決定していくことで、プロセッサの消費電力の低減を実現する。

【００１８】また、特願２０００－２８７８８３号明細書には、アプリケーションのリアルタイム要求に応えつつプロセッサによる電力消費を削減することができるマルチプロセッサ構成システムについて開示されている。すなわち、動作周波数と電源電圧をオペレーティング・システムの制御により動的に変化させることができるプロセッサを複数備えたマルチプロセッサ構成システムにおいて、各プロセッサ毎に、起動された各タスクを遅滞なく処理するために必要な動作周波数を適応的に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適な電源電圧を決定していくことで、各プロセッサ並びにシステム全体の消費電力の低減を実現する。

【００１９】しかしながら、特願２０００－２８７８８２号並びに特願２０００－２８７８８３号はいずれも、起動間隔と最大所要時間があらかじめ分っているリアルタイム タスクを対象とするものであり、ＧＵＩ処理のように、あらかじめ実行開始時間や処理時間が予測することができないタスクを実行中に、省電力スケジューリングを行うものではない。

【００２０】一般には、プロセッサは、動作周波数の増大に従って演算速度が向上する一方で、消費電力（さらには発熱量）が増大する傾向にある（前述）。また、プロセッサの動作周波数とともにその電源電圧（言い換えれば消費電力）をつり上げていかなければならない。（但し、実際には、ＬＳＩ製造プロセスの微細化によって電源電圧の上限が制限されているので、電圧を上げることによって周波数を上げることは行われない。）

【００２１】プロセッサの動作周波数と電源電圧を動的制御により変化させることが可能なシステムであれば、起動された各タスクを遅滞なく処理するために必要な動作周波数を適応的に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適な電源電圧を決定していくことで、プロセッサの消費電力を低減することが可能と思料される。

【００２２】しかしながら、ＧＵＩ処理のように、あらかじめ実行開始時間や処理時間が予測することができないタスクを実行中に、動作周波数と電源電圧の設定によりプロセッサの低消費電力化を実現した従来技術は見当たらない。

【００２３】

【発明が解決しようとする課題】本発明の目的は、応答時間に対する要求が課されたタスクを含んだ１以上のタスクを実行するプロセッサに関する優れた電力削減技術を提供することにある。

【００２４】本発明の更なる目的は、動作周波数と電源電圧を変化させる機構を備えたプロセッサに関する優れた電力削減技術を提供することにある。

【００２５】本発明の更なる目的は、ＧＵＩ（Graphical User Interface）処理のように、あらかじめ実行開始時間や処理時間が予測することができないタスクを実行時の動作周波数と電源電圧を最適に設定することにより低消費電力化を実現することができる、プロセッサに関する優れた電力削減技術を提供することにある。

【００２６】

【課題を解決するための手段及び作用】本発明は、上記課題を参酌してなされたものであり、その第１の側面は、応答時間に対する要求が課せられた処理タスクを実行可能なプロセッサを含んだ演算処理システム又は演算処理制御方法であって、前記プロセッサの稼動中の各時点において、該応答時間に対する要求を満たすために充分な前記プロセッサの動作周波数を決定する動作周波数

決定手段又はステップと、前記動作周波数決定手段又はステップによる決定結果に基づく動作周波数クロックを生成して前記プロセッサに供給するプロセッサ・クロック生成手段又はステップと、を具備することを特徴とする演算処理システム又は演算処理制御方法である。

【００２７】但し、ここで言う「システム」とは、複数の装置（又は特定の機能を実現する機能モジュール）が論理的に集合した物のことを言い、各装置や機能モジュールが単一の筐体内にあるか否かは特に問わない。

【００２８】前記動作周波数決定手段は、応答時間に対する要求が課せられた処理タスクが待ち状態にあるときは、前記プロセッサに課された他のタスクを処理するのに充分な前記プロセッサの動作周波数を決定するようにする。また、処理タスクが待ち状態を脱してから所定時間が経過したことに応じて前記プロセッサの動作周波数を段階的に増大させるようにする。

【００２９】したがって、本発明の第１の側面に係る演算処理システム又は演算処理制御方法によれば、処理タスクが待ち状態を脱したときを利用して、ユーザが感じる反応時間を増大させることなく（すなわち、ユーザの使用感に悪影響を与えることなく）、タスク実行によってプロセッサが消費する電力を好適に削減することができる。

【００３０】例えば、ＧＵＩアプリケーションの応答時間は０．１秒程度で充分である。もし、これよりも早く応答したとしても、ユーザのシステム使用感はほとんど向上しない（体感されない）。通常であれば０．１秒以下で実行できる処理を実行するときには、０．１秒程度の比較的遅い応答速度で実行されるようにプロセッサの動作周波数を制御することによって、ユーザの使用感に悪影響を及ぼすことなく、システムの消費電力を削減することができる。

【００３１】同様に、ディスク装置からのデータ読出し要求実行時においても、ディスク装置側からの完了イベントの通知を受けてからの経過時間に応じて、プロセッサの動作周波数を徐々に回復させることによって、ユーザの使用感に悪影響を与えないようにしながら、システムの消費電力を削減することができる。

【００３２】また、ネットワーク コネクションを介してデータ読み出しを依頼する場合においても、リモート装置側からの完了イベントの通知を受けてからの経過時間に応じて、プロセッサの動作周波数を徐々に回復させることによって、ユーザの使用感に悪影響を与えないようにしながら、システムの消費電力を削減することができる。

【００３３】本発明の第１の側面に係る演算処理システム又は演算処理方法は、前記動作周波数決定手段により決定された動作周波数で前記プロセッサを駆動させるために充分な前記プロセッサの電源電圧を決定する電源電圧決定手段又はステップと、前記電源電圧決定手段又はステップによる決定結果に基づく電源電圧を生成して前記プロセッサに供給するプロセッサ電源供給手段又はステップとをさらに備えていてもよい。

【００３４】一般に、プロセッサの動作周波数を増大させるためには供給電源の電圧を吊り上げる必要がある。プロセッサにおいて起動された各タスクを遅滞なく処理するために必要な動作周波数を適応的に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適な電源電圧を決定していくことで、プロセッサの消費電力を効果的に低減することが可能となる。

【００３５】また、前記動作周波数決定手段又はステップは、処理タスクが待ち状態を脱してから所定時間が経過したことに応じて前記プロセッサの動作周波数を段階的に増大させるようにしてもよい。

【００３６】あるいは、前記動作周波数決定手段又はステップは、処理タスクが待ち状態を脱してから所定時間が経過するまでの期間に前記プロセッサの動作周波数を連続的に増大させるようにしてもよい。

【００３７】また、本発明の第２の側面は、応答時間に対する要求が課せられた処理タスクを実行可能なプロセッサによる演算処理の制御をコンピュータ システム上で実行するように記述されたコンピュータ ソフトウェアをコンピュータ可読形式で物理的に格納した記憶媒体であって、前記コンピュータ ソフトウェアは、前記プロセッサの稼動中の各時点において、該応答時間に対する要求を満たすために充分な前記プロセッサの動作周波数を決定する動作周波数決定ステップと、前記動作周波数決定ステップによる決定結果に基づく動作周波数クロックを生成して前記プロセッサに供給するプロセッサ・クロック生成ステップと、を具備することを特徴とする記憶媒体である。

【００３８】本発明の第２の側面に係る記憶媒体は、例えば、様々なプログラム コードを実行可能な汎用コンピュータ システムに対して、コンピュータ・ソフトウェアをコンピュータ可読な形式で提供する媒体である。このような媒体は、例えば、ＣＤ（Compact Disc）やＦＤ（Floppy Disk）、ＭＯ（Magneto-Optical disc）などの着脱自在で可搬性の記憶媒体である。あるいは、ネットワーク（ネットワークは無線、有線の区別を問わない）などの伝送媒体などを経由してコンピュータ・ソフトウェアを特定のコンピュータ・システムに提供することも技術的に可能である。

【００３９】このような記憶媒体は、コンピュータ・システム上で所定のコンピュータ・ソフトウェアの機能を実現するための、コンピュータ・ソフトウェアと記憶媒体との構造上又は機能上の協働的関係を定義したものである。換言すれば、本発明の第２の側面に係る記憶媒体を介して所定のコンピュータ・ソフトウェアをコンピュータ・システムにインストールすることによって、コンピュータ・システム上では協働的作用が発揮され、本発

明の第１の側面に係る演算処理システム及び演算処理制御方法と同様の作用効果を得ることができる。

【００４０】本発明のさらに他の目的、特徴や利点は、後述する本発明の実施例や添付する図面に基づくより詳細な説明によって明らかになるであろう。

【００４１】

【発明の実施の形態】以下、図面を参照しながら本発明の実施例を詳解する。

【００４２】１．システム構成

図１には、本発明の実施に供される演算処理システム１０のハードウェア構成を模式的に示している。同図に示すように、演算処理システム１０は、プロセッサ１１と、ＲＡＭ（Random Access Memory）１２と、ＲＯＭ（Read Only Memory）１３と、周辺デバイス１４と、タイマ１５とを含んでいる。

【００４３】プロセッサ１１は、演算処理システム１０のメイン・コントローラであり、オペレーティング・システム（ＯＳ）の制御下で、各種のプログラム・コードを実行するようになっている。

【００４４】オペレーティング・システムがプログラム実行を管理・制御する単位は、一般に「タスク」と呼ばれる。本実施例に係るプロセッサ１１は、異なる周期で動作する複数のタスクを同時に実行するマルチタスク機構を備えているものとする。タスクには、次の周期の開始までに実行を完了させる必要がある「周期リアルタイム・タスク」と、このような実行完了時間に制約がない「非リアルタイム・タスク」とに大別することができる。非リアルタイム タスクには、応答時間に対する要求が課せられたタスク、若しくは、ＧＵＩ（Graphical User Interface）処理のように、待ち状態を含んだり、実行開始時間や処理時間が予測できない処理が含まれる。

【００４５】プロセッサ１１は、バス１６によって他の機器類（後述）と相互接続されている。バス１６上の各機器にはそれぞれ固有のメモリ アドレス又はＩ／Ｏアドレスが付与されており、プロセッサ１１はこれらアドレスを指定することによって所定の機器へのアクセスが可能となっている。バス１６は、アドレス・バス、データ・バス、コントロール・バスを含む共通信号伝送路である。

【００４６】ＲＡＭ１２は、書き込み可能なメモリであり、プロセッサ１１において実行されるプログラム・コードをロードしたり、実行プログラムの作業データを一時格納するために使用される。プログラム・コードには、例えば、ＢＩＯＳ（BasicInput/Output System：基本入出力システム）、周辺機器をハードウェア操作するためのデバイス・ドライバ、オペレーティング・システム、アプリケーションなどが挙げられる。

【００４７】ＲＯＭ１３は、所定のコードやデータを恒久的に記憶するための不揮発メモリであり、例えば、ＢＩＯＳや始動時の自己診断プログラム（Power On Self Test：ＰＯＳＴ）などを格納している。

【００４８】周辺デバイス１４には、ディスプレイやプリンタのようなユーザ出力装置、キーボードやマウスのようなユーザ出力装置、ハード・ディスクやその他のメディア・ドライブからなる外部記憶装置、ネットワーク・インターフェース・カード（ＮＩＣ）のような通信装置が含まれる。

【００４９】各周辺デバイスには、割り込みレベルが割り当てられており、所定のイベント発生（例えばキーボード入力やマウス・クリックなどのＧＵＩ処理や、ハード ディスクにおけるデータ転送の完了など）に応答して、割り込み要求信号線１９を介してプロセッサ１１に通知することができる。プロセッサ１１は、このような割り込み要求に応答して、対応する割り込みハンドラを実行する。この種の割り込み処理は、待ち状態を含んだり、実行開始時間や処理時間が予測できない処理である。

【００５０】タイマ１５は、タイマ信号を所定周期で発生させる装置である。タイマ１５にも割り込みレベルが割り当てられており、割り込み要求信号線１９を介してプロセッサ１１に対して周期的な割り込みを発生するようになっている。（但し、周期の異なる複数の周期リアルタイム・タスクが存在する場合、タイマ信号は周期的な割り込みにはならない。）

【００５１】上述したようなシステム１０の各コンポーネントには、電源装置１７からの電力が電源供給線１８を介して供給される。電源装置１７は、例えばバッテリや商用ＡＣ電源で構成されるが、ＡＣ／ＤＣアダプタやＤＣ／ＤＣコンバータによって一定の電源電圧を供給することができる。

【００５２】図示の例では、プロセッサ１１に対しては、専用のＤＣ／ＤＣ変換器２１が配設されている。本実施形態では、プロセッサ１１は、オペレーティング・システムの制御下で、プロセッサ用ＤＣ／ＤＣ変換器２１からの供給電圧を設定する機構を備えている。

【００５３】また、プロセッサ１１は、プロセッサ・クロック生成器２２が発生する動作クロックを入力して、その動作周波数に同期的に駆動する。一般には、動作周波数の増大とともに、プロセッサ１１のパフォーマンスすなわち処理速度は向上するとともに、その消費電力も増大する。本実施例では、プロセッサ１１は、オペレーティング・システムの制御下で、プロセッサ・クロック生成器２２が生成するクロックの動作周波数を設定する機構を備えている。

【００５４】なお、プロセッサ１１に対する電源電圧と動作周波数の双方をプロセッサ用ＤＣ／ＤＣ変換器２１とプロセッサ・クロック生成器２２の各々によって動的に制御する必要は必ずしもなく、いずれか一方の動作によっても本発明の効果を実現することができる。言い換

えれば、オペレーティング・システムは、プロセッサ１１の電源電圧と動作周波数の双方を動的制御するのではなく、いずれか一方のみを制御する場合であっても、後述する本発明の動作特性並びに効果を実現することができる。また、電源電圧と動作周波数のいずれか一方のみを演算処理により設定し、他方はその設定に自動的に追従するように構成してもよい（例えば、プロセッサ１１の周波数をオペレーティング・システムが設定することによって、その周波数で動作するために必要な最小の電源電圧が自動的にプロセッサ１１に供給されるように構成してもよい）。

【００５５】プロセッサ１１は、一般に、ＣＭＯＳ（Complementary Metal Oxide semiconductor：相補性金属酸化膜半導体）論理素子で構成される。ＣＭＯＳ論理素子には、その遅延時間が電源電圧によって変化するという特性がある。これは、プロセッサ１１の動作周波数を下げることによって、より低い電源電圧で回路を駆動することが可能であることを意味する。また、電源電圧を高めることによって、より高い動作周波数でプロセッサ１１を動作させることが可能であることも意味する。

【００５６】ところで、ＣＭＯＳ論理素子が消費する電力は、単位時間当りのスイッチング回数（これはプロセッサ１１の動作周波数にほぼ比例する）に比例する。しかし、プロセッサ１１の処理能力はクロック周波数にほぼ比例するため、単純に動作周波数を下げただけでは、ある所定の計算を行うために必要な電力は削減できない（何故ならば、単位時間当りの消費電力を削減することはできるが、その分だけ計算に要する時間が増大するため、結局、計算を完了させるまでに消費する電力量は変わらないことになる）。

【００５７】他方、動作周波数が下がることによって、プロセッサ１１が正常に動作可能な最低電源電圧は低下する。さらに、プロセッサ１１の消費電力は電源電圧の２乗にほぼ比例する。したがって、プロセッサ１１のクロック周波数を低下させるとともに電源電圧を低下させることによって、ある計算に要する消費電力量を削減することが可能である。

【００５８】本実施形態では、プロセッサ１１の動作クロック周波数と電源電圧は、プロセッサ１１が実行する電力マネージャ（後述）によって、システムに課せられた性能要求を満たす範囲で、可能な限り低くなるように制御される。電力マネージャは、例えば、オペレーティング システムの一部の機能として実装することができる。

【００５９】電力マネージャは、プロセッサ１１への動作クロック周波数と電源電圧のうちいずれか一方だけを設定して、他方はその設定に自動的に追従するように構成してもよい。例えば、電力マネージャがプロセッサ１１の動作周波数だけを設定することによって、その周波数で動作するために必要な最小の電源電圧が自動的に供給されるように構成してもよい。

【００６０】２．アプリケーション

本発明は、応答時間の対する要求が課された処理、あるいは、待ち状態を含んだり、実行開始時間や処理時間が予測できない処理を実行中のプロセッサ１１の消費電力を低減するように、その動作周波数や電源電圧の設定を行うものである。以下では、応答時間の対する要求が課された処理の例として、キーボードやマウスからのユーザ入力に応答して動作するＧＵＩアプリケーションをとり上げて説明する。

【００６１】ＧＵＩアプリケーションの多くは、以下に挙げるようなさまざまなイベントに逐次反応することによって処理を進めるように実装されている。すなわち、

【００６２】（１）ユーザが、キーボード内のあるキーを押下操作したとき、あるいは押下していたキーを解放した。

（２）マウス ボタンが押された、又は解放された。

（３）マウス ポインタが、ウィンドウ中のある領域内に突入した、あるいはその領域から退出した。

【００６３】図２には、ＧＵＩフレームワーク、並びにＧＵＩアプリケーションの構成を模式的に示している。

【００６４】ＧＵＩフレームワークは、キーボードやマウス、ディスプレイなどのようにＧＵＩ入出力を行う周辺デバイスを制御する。これら周辺デバイス上で、「キー入力された」、「マウス操作された」、「ディスプレイ表示が更新された」などのイベントが発生すると、ＧＵＩフレームワークは、そのイベント オブジェクトをイベント キューに送信する。

【００６５】一方、ＧＵＩアプリケーションは、起動後、アプリケーションの初期設定を行った後、イベント キューにイベント オブジェクトが到着するまで待機する。すなわち、ＧＵＩフレームワークのイベント キューに対してイベント待ち手続の呼出しを行う。

【００６６】アプリケーションが処理すべきイベント オブジェクトがイベント キューに到着すると、次いで、イベントの種類を特定して、各イベントに対応した処理を実行する。

【００６７】イベントの処理においては、必要に応じて、ＧＵＩフレームワーク内の描画部に対して表示の更新を依頼することができる。描画部は、この依頼に応答して、ディスプレイ上の該当ウィンドウの描画処理を行う。

【００６８】図２において、ＧＵＩフレームワークがイベント到着によってＧＵＩアプリケーションへ制御を移行させてから、ＧＵＩアプリケーションがイベント・キューにイベント オブジェクトが到着するまで待つ処理の実行を開始するまでの時間が、ＧＵＩアプリケーションの応答時間に相当する。

【００６９】多くのアプリケーションにおいて、ＧＵＩアプリケーションの応答時間は０．１秒程度で充分であ

る。もし、これよりも早く応答したとしても、ユーザのシステム使用感はほとんど向上しない（体感されない）。さらに、既に述べたように、プロセッサ１１の動作周波数を低下させることによって、同一の処理をより低い電力消費量で実行することができる。

【００７０】そこで、本実施形態においては、通常であれば０．１秒以下で実行できる処理を実行するときには、０．１秒程度の比較的遅い応答速度で実行されるようにプロセッサ１１の動作周波数と電源電圧を制御するようにした。このような制御を行うことによって、ユーザの使用感に悪影響を及ぼすことなく、システム１０の消費電力を削減することができる、という点を充分理解されたい。

【００７１】例えば、システム１０内でただ１つのタスクが動作しており、且つ、それがＧＵＩ処理タスクだった場合、イベントが発生してから一定の時間は、プロセッサ１１を低い動作周波数で動作させるが、その後（所定時間経過後）、高い動作周波数で制御するようにして、イベント発生に伴う待ち状態からの再開時においてユーザの使用感に悪影響を及ぼすことなく、プロセッサ１１の消費電力を削減することができる。

【００７２】図３には、ＧＵＩ処理タスク実行中において、イベント発生時におけるプロセッサ１１の動作周波数の制御タイミング例を示している。同図に示す例では、プロセッサ１１は、フル稼働するために必要な高レベル要求値と、比較的遅い応答速度を満たすために最低限必要な低レベル要求値という、２段階の動作周波数で駆動することができるものとする。

【００７３】まず、ＧＵＩアプリケーションがユーザからのＧＵＩ入力を求めるイベント待ち状態に変位すると、プロセッサ１１の動作周波数を停止して、最大限に低消費電力化を図る。

【００７４】次いで、ユーザによるＧＵＩ入力が行われ、イベントがＧＵＩアプリケーションに到着すると、待ち状態から脱して、一定期間（要求レベル変更時間）は、比較的応答速度が遅い動作でもユーザの使用感に悪影響を与えない。そこで、プロセッサ１１に対して低レベル要求値の動作周波数を供給して、低消費電力化を図る。

【００７５】そして、要求レベル変更時間が経過すると、応答速度が遅いままではユーザの使用感に悪影響を及ぼし始めるので、プロセッサ１１に対して高レベル要求値の動作周波数を供給して、最速の応答速度に戻す。

【００７６】プロセッサの動作周波数を低下させた場合、従来のシステムで長い応答時間を要していた処理は、さらに応答時間が長くなるという問題が生じる。これに対し、本実施形態では、要求レベル変更時間を例えば０．１秒程度に設定することにより、応答の遅れは約０．１秒以下になる。０．１秒の性能劣化は、ユーザの使用感にほとんど悪影響を与えないので、問題とはならない。

【００７７】図３に示す例では、システム１０がイベント待ち状態を脱出してからただ一度だけプロセッサ１１の動作周波数を変化させている。しかしながら、各処理の計算時間が広くばらついているような場合には、動作周波数が一段階の変化しかしなければ、満足の行く省電力制御を実現できないことも考えられる。このような場合のために、ＧＵＩアプリケーションへのイベント到着からの経過時間に応じて多段階でプロセッサ１１の動作周波数を変化させることも有効であると思料される。

【００７８】図４には、ＧＵＩ処理タスク実行中において、イベント到着からの経過時間に応じて多段階でプロセッサ１１の動作周波数を変化させる場合の制御タイミング例を示している。同図に示す例では、プロセッサ１１は、動作周波数が低い方から順に、要求値レベル１、要求値レベル２、要求値レベル３という３段階のの動作周波数で駆動することができるものとする。

【００７９】まず、ＧＵＩアプリケーションがユーザからのＧＵＩ入力を求めるイベント待ち状態に変位すると、プロセッサ１１の動作周波数を停止して、最大限に低消費電力化を図る。

【００８０】次いで、ユーザによるＧＵＩ入力が行われ、イベントがＧＵＩアプリケーションに到着すると、待ち状態から脱して、最初の要求レベル変更時間までの期間は、要求値レベル１に相当する動作周波数をプロセッサ１１に供給して、応答速度が最も遅い状態で動作させる。

【００８１】次いで、最初の要求レベル変更時間が経過すると、要求値レベル２に相当する動作周波数をプロセッサ１１に供給して、応答速度を少し高める。

【００８２】さらに、２回目の要求レベル変更し時間が経過すると、要求値レベル３に相当する動作周波数をプロセッサ１１に供給して、最速の応答速度に戻す。

【００８３】図３及び図４に示す例では、システム１０がイベント待ち状態を脱出してからプロセッサ１１の動作周波数を段階的に変化させているが、イベント到着から所定時間が経過するまでに動作周波数を最大に戻すまでの期間に動作周波数を変化させる態様はこれらに限定されるものではない。例えば、イベント到着から所定時間が経過するのでの間に、動作周波数を変化させる段階を無限に増やし、連続的に変化させることも有効である。

【００８４】図５には、ＧＵＩ処理タスク実行中において、イベント到着からの経過時間に応じてプロセッサ１１の動作周波数を連続的に変化させる場合の制御タイミング例を示している。同図に示す例では、プロセッサ１１は、動作周波数を連続的に変化させて駆動することができるものとする。

【００８５】まず、ＧＵＩアプリケーションがユーザからのＧＵＩ入力を求めるイベント待ち状態に変位する

と、プロセッサ１１の動作周波数を停止して、最大限に低消費電力化を図る。

【００８６】次いで、ユーザによるＧＵＩ入力が行われ、イベントがＧＵＩアプリケーションに到着すると、待ち状態から脱して、最初の要求レベル変更時間までの期間は、プロセッサ１１の動作周波数を要求レベル１まで連続的に変化させる。

【００８７】次いで、最初の要求レベル変更時間から２回目の要求レベル変更時間までの期間中は、プロセッサ１１の動作周波数を要求レベル１から要求レベル２まで連続的に変化させる。

【００８８】そして、２回目の要求レベル変更器官を経過した後は、所定時間内に、プロセッサ１１の動作周波数を要求レベル２から要求レベル３まで、連続的に変化させて、最速の応答速度に戻す。

【００８９】３．他の実施形態

これまでは、ＧＵＩアプリケーション実行時を例にとって、プロセッサ１１の省電力制御について説明してきたが、本発明の要旨は、ＧＵＩアプリケーションへの適用に限定されるものではなく、一般に、応答時間に対する要求があり、且つ、あるイベントに反応して処理を行うアプリケーション全般に対して適用することができる。

【００９０】本発明の他の実施形態として、ハード ディスクなどのディスク装置（あるいはその他のタイプの外部記憶装置）からのデータ到着待ち状態における適用例を挙げることができる。

【００９１】プロセッサ１１がディスク装置から要求データが到着するまで待ち状態に陥り、その後、転送データを基に画面表示を行う処理を繰り返すアプリケーションを実行する場合を例にとって、図６を参照しながら説明する。

【００９２】この場合、アプリケーション プログラムは、オペレーティング システム（又は、ＯＳ内のファイル マネージャ）に対して、ディスク装置からのデータ読出しを依頼する。

【００９３】オペレーティング システム（又はファイル マネージャ）は、該当するディスク装置に対してデータ読出し命令を発行する。オペレーティング システムとディスク装置の間には所定のディスク ドライバ（図示しない）などのソフトウェアが介在していてもよい。

【００９４】このデータ読出し要求を発行した後、プロセッサ１１はイベント待ち状態に陥り、動作周波数を停止させる。

【００９５】その後、ディスク装置が要求されたデータの読出しを完了すると、オペレーティング システムに対して完了イベントを通知する。オペレーティング システムは、さらに要求元アプリケーションに対して完了イベントを通知する。

【００９６】この完了イベントは、ＧＵＩアプリケーションへの適用例における「イベント到着」に相当する。したがって、この完了イベントの通知を受けてからの経過時間に応じて、例えば図３～図５に示したような形式で、プロセッサ１１の動作周波数を徐々に回復させることができる。この結果、ユーザの使用感に悪影響を与えないようにしながら、システム１０の消費電力を削減することができる。

【００９７】アプリケーション プログラムは、オペレーティング システムを経由して受け取った読出しデータを基に、ディスプレイに読出し結果を表示することができる。この際、ＧＵＩフレームワーク（前述）が介在してもよい。

【００９８】また、本発明の他の実施形態として、ネットワーク コネクションを介してデータ入出力を行うアプリケーションに対する適用例を挙げることができる。

【００９９】プロセッサ１１がネットワーク コネクションを介して要求データが到着するまで待ち状態に陥り、その後、転送データを基に画面表示を行う処理を繰り返すアプリケーションを実行する場合を例にとって、図７を参照しながら説明する。

【０１００】この場合、アプリケーション プログラムは、オペレーティング システム（又は、通信プロトコル ソフトウェア）に対して、ネットワーク コネクションを介した（リモート ディスクからの）データ読出しを依頼する。

【０１０１】オペレーティング システム（又は通信プロトコル ソフトウェア）は、該当するリモート ディスクに対してデータ読出し命令を発行する。

【０１０２】このデータ読出し要求を発行した後、プロセッサ１１はイベント待ち状態に陥り、動作周波数を停止させる。

【０１０３】その後、リモート ディスクが要求されたデータの読出しを完了すると、オペレーティング システムに対して完了イベントを通知する。オペレーティング システムは、さらに要求元アプリケーションに対して完了イベントを通知する。

【０１０４】この完了イベントは、ＧＵＩアプリケーションへの適用例における「イベント到着」に相当する。したがって、この完了イベントの通知を受けてからの経過時間に応じて、例えば図３～図５に示したような形式で、プロセッサ１１の動作周波数を徐々に回復させることができる。この結果、ユーザの使用感に悪影響を与えないようにしながら、システム１０の消費電力を削減することができる。

【０１０５】アプリケーション プログラムは、オペレーティング システムを経由して受け取った読出しデータを基に、ディスプレイに読出し結果を表示することができる。この際、ＧＵＩフレームワーク（前述）が介在してもよい。また、アプリケーションは、ネットワーク経由で、データ読出し先に結果を送信する。

【0106】4．ソフトウェア構成
図3～図5に示したようなプロセッサ11の動作周波数制御による電力管理は、例えば、オペレーティング システムの1つの機能として実装することができる。以下では、オペレーティング システム内の当該機能モジュールのことを、「電力マネージャ」と呼ぶことにする。
【0107】図8には、図3に示したように、イベント到着以後1段階でプロセッサ11の動作周波数を回復させる場合の電力マネージャのソフトウェア構成を模式的に示している。
【0108】図8に示すように、電力マネージャは、各タスク毎に1つずつ、フレームワーク部とタイマを用意する。各タスクは、以下の3種類の変数を保持している。これらの変数の持つ意味は、図3を参照しながら既に説明した通りである。
【0109】（1）低レベル要求値（値域は［0，1．0］）
（2）高レベル要求値（値域は［0，1．0］）
（3）要求レベル変更時間
【0110】図8に示す例では、フレームワーク部は、以下の手順で動作する。
【0111】ステップ1：タスクが「イベント キューにイベント オブジェクトが到着するまで待つ」処理の実行開始をフレームワーク部に依頼したとき（手順P1）、電力マネージャに対して、このタスクがプロセッサ11の動作周波数に関してまったく要求を行っていないことを通知する（手順P2）。また、タイマが動作中だった場合には、それを停止させる。
【0112】ステップ2：イベント キューにイベントが到着したとき（手順P3）、タスクがプロセッサ11の動作周波数に対して低レベルな要求を行っていることを通知して（手順P4）、タイマを始動させる（手順P5）。また、対応するアプリケーションにイベントの到着を通知する（手順P6）。タイマは、始動してから要求レベル変更時間が経過したときに、電力マネージャに対して、このタスクが高レベルな要求を行っていることを通知する（手順P7）。
【0113】また、図9には、図4に示したように、イベント到着以後1段階でプロセッサ11の動作周波数を回復させる場合のソフトウェア構成を模式的に示している。
【0114】図9に示すように、電力マネージャは、各タスク毎に以下の4種類の変数を用意する。これらの変数の意味は、以下に示す通りである。
【0115】（1）要求値段階番号
現在、プロセッサ11の動作周波数が何段階目に達しているかを保持する変数である。例えば、図4に示す例では、「2．イベントの到着」直後から「3．最初の要求値変更時間が経過」直前までは、この変数の値は1である。その後、要求値の変更が行われる度に、要求値段階番号は1ずつ加算される。また、要求が発生していない（すなわち、タスクがイベント待ち状態にある）ときは、値0をとる。
【0116】（2）要求値配列
要求値配列のn番目の要素には、変数「要求値段階番号」が値nをとるときの要求値が格納されている。
【0117】（3）要求値変更時間配列
要求値変更時間配列のn番目の要素には、変数「要求値段階番号」が値nをとり、電力マネージャに対して要求「要求値配列［n］」が設定されている期間の長さが格納されている。この時間が経過したならば、次の段階の要求値が設定されることになる。但し、以下に述べる変数「段階数」を越えてnが増加することはない。
【0118】（4）段階数
段階数を保持する変数である。
【0119】図9に示す例では、フレームワーク部は、以下の手順で動作する。
【0120】ステップ1：タスクが「イベント キューにイベント オブジェクトが到着するまで待つ」処理の実行開始をフレームワーク部に依頼したとき（手順P11）、電力マネージャに対して、このタスクがプロセッサ11の動作周波数に関してまったく要求を行っていないことを通知する（手順P12）。また、タイマが動作中だった場合には、それを停止させる。さらに、フレームワーク部の要求値段階番号に0を代入する。
【0121】ステップ2：イベント キューにイベントが到着したとき（手順P13）、要求値段階番号に1を代入する（又は、1だけ増分する）。そして、第1段階（又は、次の段階）の要求値を電力マネージャに設定して（手順P14）、タイマを始動させる（手順P15）。また、対応するアプリケーションにイベントの到着を通知する（手順P16）。タイマは、要求値変更時間配列の先頭の要素が示す時間が経過したとき、電力マネージャに対して、このタスクが次の段階の要求を行っていることを通知する（手順P17）。これによって、要求値段階番号の更新と、この段階番号に対応する要求値の設定、及び、タイマの再設定が行われる。
【0122】5．電力マネージャ
マルチタスク環境下でも本発明を好適に実現するために、電力マネージャは、複数のタスクの性能要求を統合し、プロセッサ11の適切な動作周波数、及び電源電圧を決定する必要がある。この項では、その決定処理を行う方法について説明する。
【0123】本実施形態では、プロセッサ11は、以下の3種類のタスクを扱うことができるものとする。
【0124】（1）リアルタイム タスク
リアルタイム タスクは、ある定められた周期で起動され、且つ、次の周期の開始までに実行を完了させる必要があるタスクのことである。但し、各リアルタイム・タスクの周期は区々である。

【０１２５】（２）高応答性タスク
応答時間の対する要求が課された処理、あるいは、待ち状態を含んだり、実行開始時間や処理時間が予測できない処理を行うタスクである。高応答性タスクは、キーボードやマウスなどのユーザ入力操作をはじめとしたさまざまなイベントに応答して実行される。
【０１２６】（３）その他のタスク
【０１２７】本実施形態では、電力マネージャは、動作中のすべてのリアルタイム タスクと、その他のタスクを考慮して、負荷の変化に応じて、プロセッサ１１の適切な動作周波数を常に計算する。この処理自体は、例えば、本出願人に既に譲渡されている特願２０００－２８７８８２号明細書に開示されている（起動された周期リアルタイム・タスク並びに非リアルタイム・タスクを遅滞なく処理するために必要なプロセッサの動作周波数を適応的に変化させるとともに、時々刻々と切り替わる動作周波数に応じて最適なプロセッサ用電源電圧を決定していくことで、プロセッサの消費電力を低減する）。そして、その計算結果が変数「リアルタイム タスク、その他のタスクの要求値」に格納される。
【０１２８】リアルタイム タスクは、その性質上、他のタスクよりも優先的に実行する必要がある。すなわち、実行可能なリアルタイム タスクが存在するときには、必ずそれらが実行されることが保証されているものとする。
【０１２９】動作中の高応答性タスクの要求値は、フレームワーク部を経由して電力マネージャへと通知される（例えば、図８中の手順Ｐ２，Ｐ４，Ｐ７）。この通知が行われる度に（すなわち、高応答性タスクの要求が変化する度に）、電力マネージャは以下の値を計算して、これをプロセッサ１１の動作周波数として設定する。
【０１３０】
【数１】

$$\max\left\{f_{max}\cdot\min\left\{1.0,\sum_{g\in G}r_g\right\},f_{RT}\right\}$$

但し、

f_{max}：最大動作周波数
G：高応答性タスク
r_g：高応答性タスクgの要求値
f_{RT}：リアルタイム・タスクとその他のタスクの負荷を基に決定された動作周波数

【０１３１】［追補］以上、特定の実施例を参照しながら、本発明について詳解してきた。しかしながら、本発明の要旨を逸脱しない範囲で当業者が該実施例の修正や代用を成し得ることは自明である。すなわち、例示という形態で本発明を開示してきたのであり、限定的に解釈されるべきではない。本発明の要旨を判断するためには、冒頭に記載した特許請求の範囲の欄を参酌すべきである。
【０１３２】
【発明の効果】以上詳記したように、本発明によれば、応答時間に対する要求が課されたタスクを含んだ１以上のタスクを実行するプロセッサに関する優れた電力削減技術を提供することができる。
【０１３３】また、本発明によれば、動作周波数と電源電圧を変化させる機構を備えたプロセッサに関する優れた電力削減技術を提供することができる。
【０１３４】また、本発明によれば、ＧＵＩ（Graphical User Interface）処理のように、あらかじめ実行開始時間や処理時間が予測することができないタスクを実行時の動作周波数と電源電圧を最適に設定することにより低消費電力化を実現することができる、プロセッサに関する優れた電力削減技術を提供することができる。
【０１３５】また、本発明によれば、ユーザが感じる反応時間を増大させることなく（すなわち、ユーザの使用感に悪影響を与えることなく）、ＧＵＩアプリケーションやネットワーク アプリケーションなどの実行によってプロセッサが消費する電力を好適に削減することができる。また、プロセッサの消費電力が低減されることにより、その発熱量も抑制することができる。
【図面の簡単な説明】
【図１】本発明の実施に供される演算処理システム１０のハードウェア構成を模式的に示した図である。
【図２】ＧＵＩフレームワーク、並びにＧＵＩアプリケーションの構成を模式的に示した図である。
【図３】ＧＵＩ処理タスク実行中において、イベント発生時におけるプロセッサ１１の動作周波数の制御タイミング例を示した図である。
【図４】ＧＵＩ処理タスク実行中において、イベント発生時におけるプロセッサ１１の動作周波数の制御タイミングの他の例を示した図である。
【図５】ＧＵＩ処理タスク実行中において、イベント発生時におけるプロセッサ１１の動作周波数の制御タイミングの他の例を示した図である。
【図６】ディスク装置から要求データが到着するまで待

ち状態に陥り、その後、転送データを基に画面表示を行う処理を繰り返すアプリケーションを実行する場合における、本発明の適用例を模式的に示した図である。

【図７】ネットワーク コネクションを介してデータ入出力を行うアプリケーションに対する本発明の適用例を模式的に示した図である。

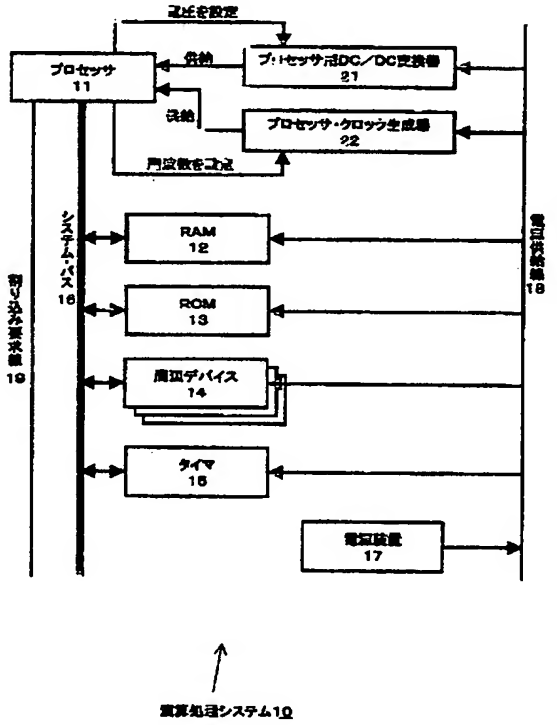
【図８】図３に示したように、イベント到着以後、１段階でプロセッサ１１の動作周波数を回復させる場合の電力マネージャのソフトウェア構成を模式的に示した図である。

【図９】図４に示したように、イベント到着以後、多段階でプロセッサ１１の動作周波数を回復させる場合の電力マネージャのソフトウェア構成を模式的に示した図である。

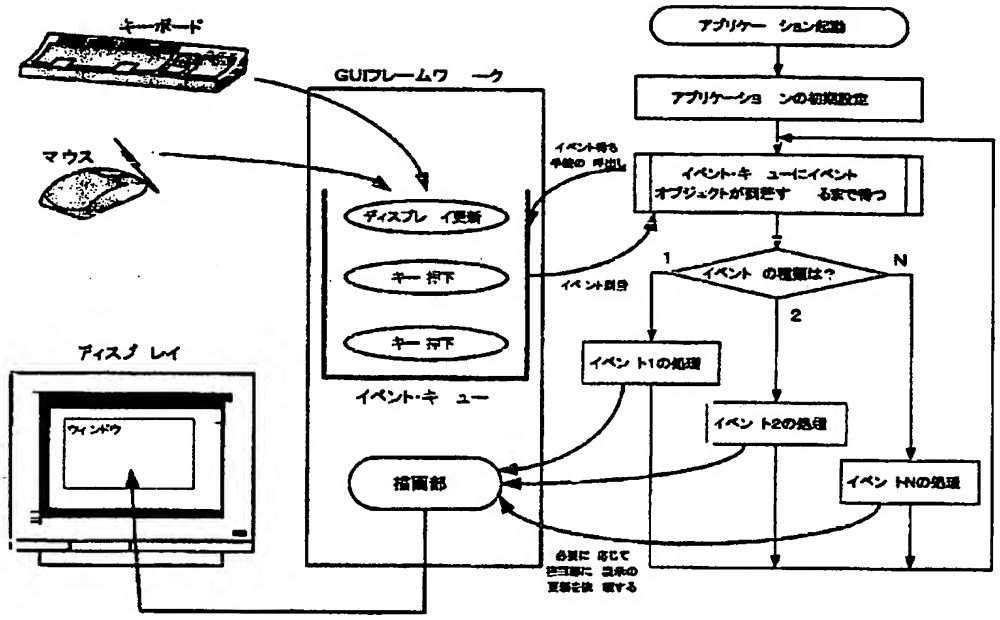
【符号の説明】

１０…演算処理システム
１１…プロセッサ
１２…ＲＡＭ
１３…ＲＯＭ
１４…周辺デバイス
１５…タイマ
１６…システム・バス
１７…電源装置
１８…電源供給線
１９…割り込み要求線
２１…プロセッサ用ＤＣ／ＤＣ変換器
２２…プロセッサ・クロック生成器

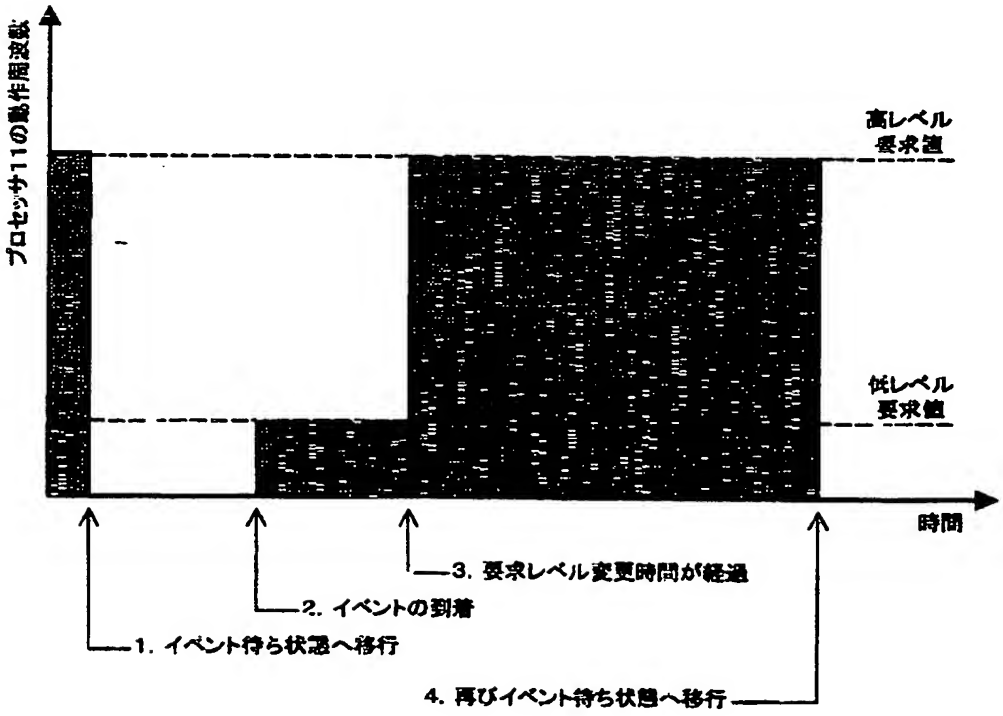
【図１】



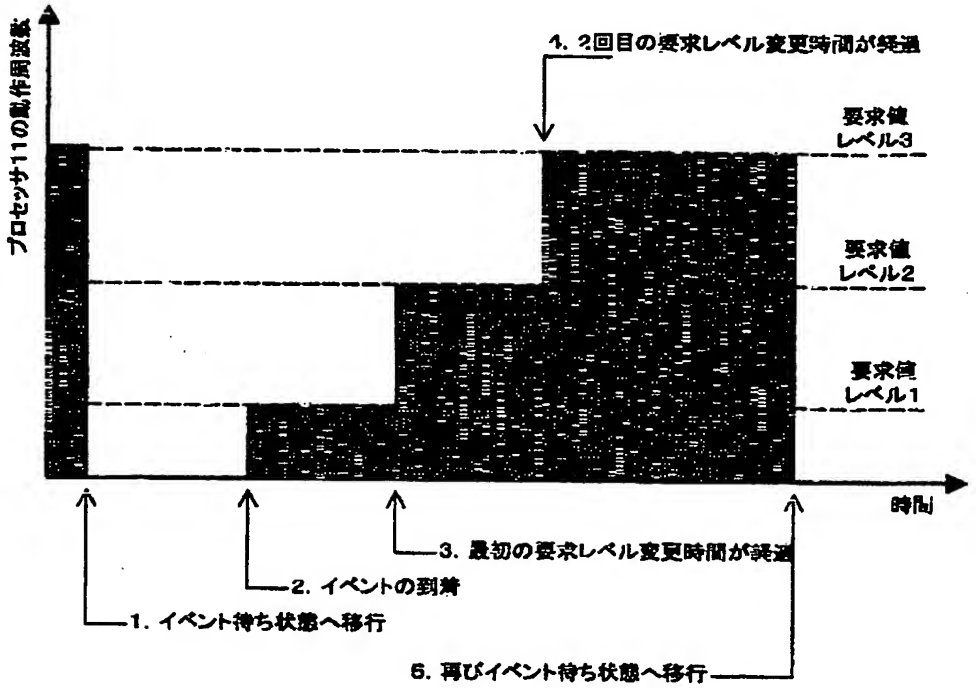
【図２】


キーボード
マウス
GUIフレームワーク
ディスプレイ更新
キー押下
キー押下
イベント・キュー
ディスプレイ
ウィンドウ
描画部
アプリケーション起動
アプリケーションの初期設定
イベント・キューにイベントオブジェクトが到着するまで待つ
イベントの種類は？
1
2
N
イベント1の処理
イベント2の処理
イベントNの処理

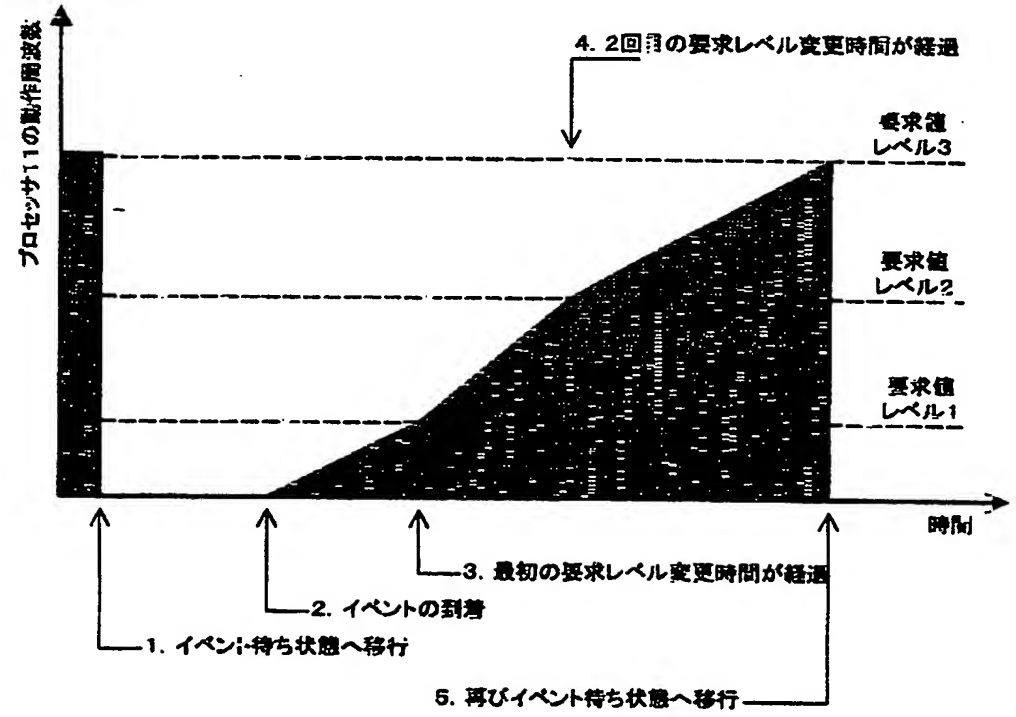
【図３】


プロセッサ11の動作周波数
高レベル要求値
低レベル要求値
時間
1. イベント待ち状態へ移行
2. イベントの到着
3. 要求レベル変更時間が経過
4. 再びイベント待ち状態へ移行

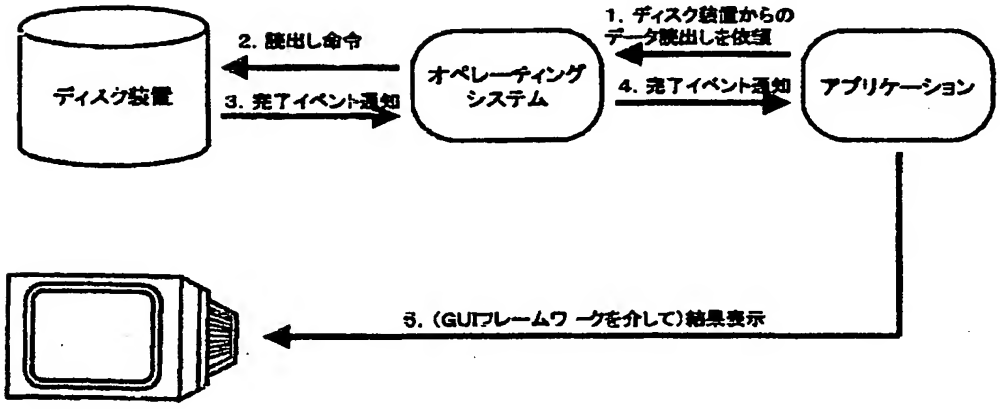
【図４】


プロセッサ11の動作周波数
4. 2回目の要求レベル変更時間が経過
要求値
レベル3
要求値
レベル2
要求値
レベル1
時間
3. 最初の要求レベル変更時間が経過
2. イベントの到着
1. イベント待ち状態へ移行
5. 再びイベント待ち状態へ移行

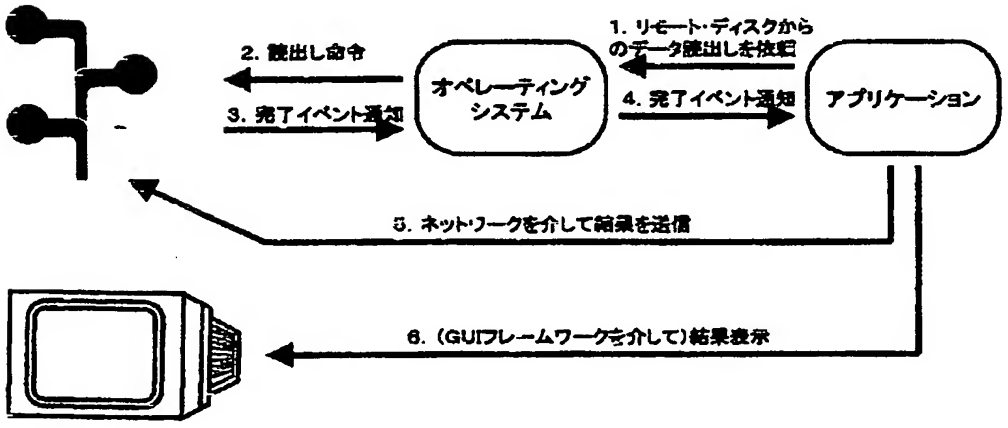
【図５】


プロセッサ11の動作周波数
4. 2回目の要求レベル変更時間が経過
要求値
レベル3
要求値
レベル2
要求値
レベル1
時間
3. 最初の要求レベル変更時間が経過
2. イベントの到着
1. イベント待ち状態へ移行
5. 再びイベント待ち状態へ移行

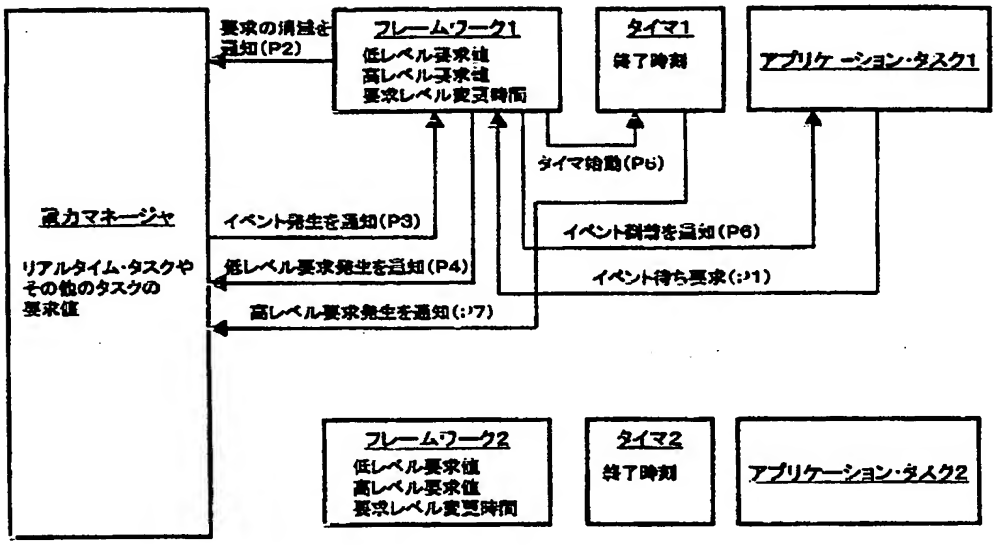
【図6】


ディスク装置
2. 読出し命令
3. 完了イベント通知
オペレーティング
システム
1. ディスク装置からの
データ読出しを依頼
4. 完了イベント通知
アプリケーション
5. (GUIフレームワークを介して)結果表示

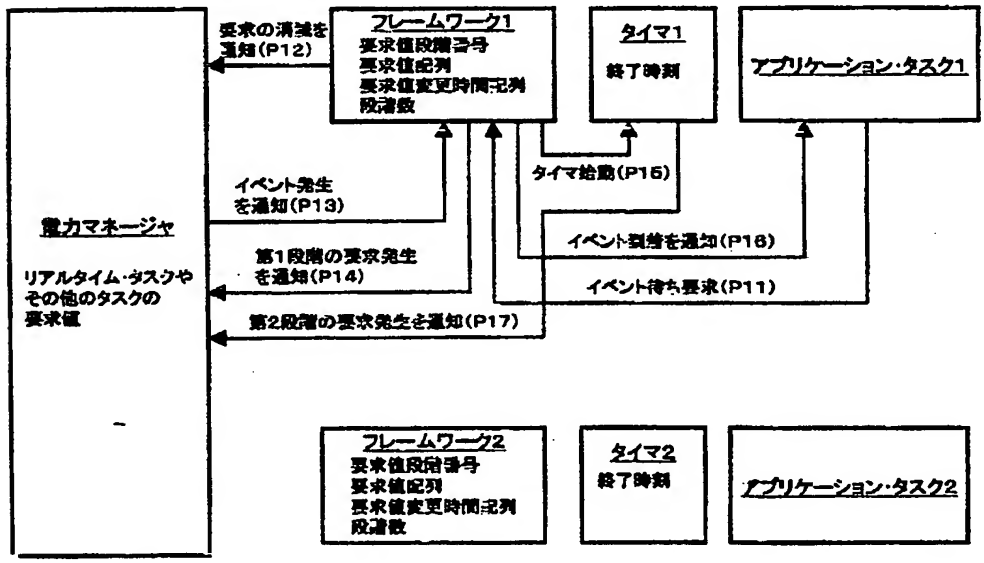
【図7】


2. 読出し命令
3. 完了イベント通知
オペレーティング
システム
1. リモート・ディスクから
のデータ読出しを依頼
4. 完了イベント通知
アプリケーション
5. ネットワークを介して結果を送信
6. (GUIフレームワークを介して)結果表示

【図8】



【図9】



フロントページの続き

Fターム(参考) 5B011 EA01 LL02 LL13
5B033 AA06 AA15 BC01
5B079 BA01 BC01
5B098 FF03 GA02 GA04